



**TECHNISCHE
UNIVERSITÄT
DRESDEN**

Technische Universität Dresden
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS-Report

**Ontology-Mediated Query Answering for Probabilistic
Temporal Data with \mathcal{EL} Ontologies (Extended Version)**

Patrick Koopmann

LTCS-Report 18-07

Postal Address:
Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
01062 Dresden

<http://lat.inf.tu-dresden.de>

Visiting Address:
Nöthnitzer Str. 46
Dresden

Contents

1	Introduction	2
2	Preliminaries	4
3	Temporal Probabilistic Knowledge Bases and Queries	4
4	Lower Complexity Bound	7
5	Upper Complexity Bound	8
6	Removing Negation	9
7	Conclusion	10
A	Appendix	12
A.1	Lower Bounds	12
A.2	Upper Bound	14
A.3	Removing Negation	17

Ontology-Mediated Query Answering for Probabilistic Temporal Data with \mathcal{EL} Ontologies (Extended Version)*

Patrick Koopmann

January 9, 2019

Abstract

Especially in the field of stream reasoning, there is an increased interest in reasoning about temporal data in order to detect situations of interest or complex events. Ontologies have been proved a useful way to infer missing information from incomplete data, or simply to allow for a higher order vocabulary to be used in the event descriptions. Motivated by this, ontology-based temporal query answering has been proposed as a means for the recognition of situations and complex events. But often, the data to be processed do not only contain temporal information, but also probabilistic information, for example because of uncertain sensor measurements. While there has been a plethora of research on ontology-based temporal query answering, only little is known so far about querying temporal probabilistic data using ontologies. This work addresses this problem by introducing a temporal query language that extends a well-investigated temporal query language with probability operators, and investigating the complexity of answering queries using this query language together with ontologies formulated in the description logic \mathcal{EL} .

1 Introduction

Ontology-mediated query answering (OMQA) recently attracted considerable attention as a technique to query incomplete data. In OMQA, queries are evaluated with respect to an ontology, which specifies background knowledge about the current domain using a formal language such as a description logic (DL), so that, using reasoning procedures, also implicit information can be queried from the data. In the standard OMQA setting, the data to be queried is assumed to be both static and precise. However, a lot of applications encounter situations where this assumption fails, yet using ontologies could prove useful. The internet has become highly dynamic, with information being frequently added and changed, and new data being generated from a variety of sources. In addition, new technologies such as smart phones and the internet of things (IoT) frequently encounter a data environment that is constantly changing. To make use of these data, there has been an increasing interest in investigating semantic and reasoning techniques that process not only static data, but streams of data, such as in the semantic stream reasoning paradigm [27]. As [27] illustrate, frequently, the data encountered in stream reasoning applications is not only temporal, but also probabilistic in nature.

As an example, consider a health or fitness monitoring application, for which one may want to use concepts from a medical ontology such as SNOMED CT [17] or Galen [28] to describe information about the health status of a patient. Specifically, such an application could be used

*Supported by the DFG within the collaborative research center SFB 912 (HAEC).

on a smartphone in combination with a sensor that measures the diastolic blood pressure of the patient while he is exercising [24]. As the sensor might be imprecise in its measurements, it might report information about whether the blood pressure of the patient is high with an associated probability, and provide this information to the application in regular time intervals. If a too high blood pressure was observed for several times during a short period, the app should give a warning to the patient, and advise him to take a break from his exercise.

In order to properly take both the temporal and the probabilistic aspects into account when querying streams of data, we propose a query language for OMQA that comes with both temporal and probabilistic operators. For this, we assume a representation of the data in form of a sequence of probabilistic data sets, which may have been obtained using further preprocessing and windowing operations. An ontology expressed in a description logic (DL) gives additional background information about the domain to be queried, so that implicit information can be queried from incomplete data through reasoning. In the above scenario, the following query could for example be used to detect whether the patients blood pressure was at least twice recorded as high during the last 10 minutes.

$$P_{>.8}(\bigcirc^{-10}\diamond(\text{HighBloodPressure}(x) \wedge \bigcirc\diamond\text{HighBloodPressure}(x)))$$

While there has been a lot of research on querying temporal data [1] and probabilistic data [7, 23] using ontologies, we are not aware of any research where both aspects are combined in the specific setting we described. In this work, we focus on the setting where the ontology is formulated in \mathcal{EL} , a DL that is known for its good computational properties, such as polynomial decidability for most common reasoning problems. This DL, which underlies the OWL EL profile of the web ontology language standard OWL, is used for many large scale ontologies, especially in the bio-medical domain and for the semantic web, such as for the ontologies SNOMED CT and Galen mentioned above. However, our hardness results already apply for simpler description logics such as *DL-Lite*, as well as for the case where no ontology is used.

Related Work Our language is an extension of the temporal query language investigated in [3, 8], which extends conjunctive queries with LTL operators. Other authors considered using these operators also as part of the DL, either to describe temporal concepts [20], or to make the axioms of the ontology itself temporal [4]. Recently, this work has been extended also to metric temporal logics, in which temporal operators are annotated with numerical time intervals [2, 12, 22]. Temporal reasoning for streams of data has recently also been considered in the context of datalog [29]. Surveys on temporal reasoning and query answering with ontologies can be found in [1, 26].

Our probabilistic query-answering framework is based on the OMQA framework for probabilistic data presented in [23]. Since this publication, several authors investigated OMQA in similar settings [7, 6, 15]. To our knowledge, the only work that combines both temporal and probabilistic query answering in the presence of description logic ontologies is [14]. Albeit, the authors consider a different setting, in which the flow of time is modelled by a Markov-process. In contrast, we we consider temporal data that are provided as a sequence of probabilistic ABoxes. In addition to settings based on probabilistic databases, there is also research on extending DLs with probability operators, such as in $\mathbf{P}\text{-SHIF}(\mathcal{D})/\mathbf{P}\text{-SHOIN}(\mathcal{D})$ [25] or $\text{Prob-ACC}/\text{Prob-EL}$ [21]. While our DL does not support probability operators, the probability operator used in our query language syntactically and semantically corresponds to the probability operator in Prob-ACC and Prob-EL .

Formal details and proofs can be found in the appendix.

2 Preliminaries

We recall the DL \mathcal{EL} [5] studied in this paper. Let $\mathbf{N}_C, \mathbf{N}_R$ and \mathbf{N}_I be countably infinite and pair-wise disjoint sets of respectively *concept names*, *role names* and *individual names*. An \mathcal{EL} *concept* is of one of the forms

$$\top \mid A \mid C_1 \sqcap C_2 \mid \exists r.C$$

where $A \in \mathbf{N}_C$, $r \in \mathbf{N}_R$, and C_1, C_2 and C are \mathcal{EL} concepts. An \mathcal{EL} *axiom* is of the form $C_1 \sqsubseteq C_2$, where C_1 and C_2 are \mathcal{EL} concepts. An \mathcal{EL} *TBox* is a set of \mathcal{EL} axioms. An *ABox* \mathcal{A} is a set of *assertions* of the form $A(a)$ and $r(a, b)$, where $A \in \mathbf{N}_C$, $r \in \mathbf{N}_R$ and $a, b \in \mathbf{N}_I$. An \mathcal{EL} knowledge base (KB) is a tuple $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where \mathcal{A} is an ABox and \mathcal{T} an \mathcal{EL} TBox.

The semantics of KBs is defined in terms of *interpretations*, which are tuples $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, $\Delta^{\mathcal{I}}$ being a set of *domain elements*, and $\cdot^{\mathcal{I}}$ an *interpretation function* that maps each $a \in \mathbf{N}_I$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ s.t. for $a \neq b \in \mathbf{N}_I$, $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ (unique name assumption, UNA), each $A \in \mathbf{N}_C$ to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, each $r \in \mathbf{N}_R$ to a subset $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function $\cdot^{\mathcal{I}}$ is extended to concepts and roles as follows:

$$\begin{aligned} \top^{\mathcal{I}} &= \Delta^{\mathcal{I}} & (C_1 \sqcap C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} \\ (\exists r.C_1)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \exists (d, e) \in r^{\mathcal{I}}, e \in C_1^{\mathcal{I}}\}, \end{aligned}$$

where C_1, C_2 are concepts and $r \in \mathbf{N}_R$. An interpretation is a model of a KB $\langle \mathcal{T}, \mathcal{A} \rangle$ (of an TBox) if for every $C \sqsubseteq D \in \mathcal{T}$, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, for every $A(a) \in \mathcal{A}$, $a^{\mathcal{I}} \in A^{\mathcal{I}}$, and for every $r(a, b) \in \mathcal{A}$, $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$.

A *conjunctive query* (CQ) takes the form $q = \exists \vec{y}. \phi(\vec{x}, \vec{y})$, where \vec{x}, \vec{y} are vectors of variables and $\phi(\vec{x}, \vec{y})$ is a conjunction over atoms of the forms $A(t_1)$ and $r(t_1, t_2)$, where $A \in \mathbf{N}_C$, $r \in \mathbf{N}_R$, and t_1 and t_2 are *terms* taken from \mathbf{N}_I , \vec{x} or \vec{y} . \vec{y} are the *answer variables* of q . Given an interpretation \mathcal{I} and a CQ q with answer variables x_1, \dots, x_n , the vector $a_1 \dots a_n \subseteq \mathbf{N}_I^n$ is an *answer of q in \mathcal{I}* if there exists a mapping $\pi : \text{term}(q) \rightarrow \Delta^{\mathcal{I}}$ s.t. $\pi(x_i) = a_i$ for $i \in [1, n]$ $\pi(b) = b^{\mathcal{I}}$ for $b \in \mathbf{N}_I$, $\pi(t) \in A^{\mathcal{I}}$ for every $A(t)$ in q , and $\langle \pi(t_1), \pi(t_2) \rangle \in r^{\mathcal{I}}$ for every $r(t_1, t_2)$ in q . $a_1 \dots a_n$ is a *certain answer* of q in a KB \mathcal{K} if it is an answer in every model of \mathcal{K} . If a query does not contain any answer variables, it is a *Boolean CQ*, and we say it is *entailed* by a KB \mathcal{K} (interpretation \mathcal{I}) if it has the empty vector as answer.

3 Temporal Probabilistic Knowledge Bases and Queries

We introduce temporal probabilistic knowledge bases (TPKBs) and temporal probabilistic queries (TPQs).

Temporal Probabilistic Knowledge Bases. Probabilistic information about a single time point is represented using a probabilistic ABox as introduced in [23]. For simplicity, we focus on assertion-independent probabilistic ABoxes (ipABoxes), though all results should easily extend to the more general case. ipABoxes are the ABox equivalent of tuple-independent probabilistic databases [16]. An *ipABox* is a set of *probabilistic ABox assertions* of the form $\alpha : p$, where α is an ABox assertion and $p \in [0, 1]$. Intuitively, $\alpha : p$ describes that the assertion α holds with a probability of at least p . Instead of $\alpha : 1$, we may just write α if the meaning is clear from the context. ipABoxes only specify a *lower bound* on the probability, to conform with the open-world semantics common in ontology-based representations.¹

¹ Note that this is different to the open-world semantics for probabilistic databases suggested in [13], which assumes a fixed upper probability for facts absent in the data.

$\Omega_{\mathcal{K}}$	\mathcal{A}'_1	\mathcal{A}'_2	\mathcal{A}'_3	\mathcal{A}'_4	\mathcal{A}'_5	$\mu_{\mathcal{K}}$
w_1	{hasBP(p, b), HighBP(b)}	\emptyset	{HighBP(b)}	{HighBP(b)}	\emptyset	0.378
w_2	{hasBP(p, b)}	\emptyset	{HighBP(b)}	{HighBP(b)}	\emptyset	0.162
w_3	{hasBP(p, b), HighBP(b)}	\emptyset	\emptyset	{HighBP(b)}	\emptyset	0.042
w_4	{hasBP(p, b)}	\emptyset	\emptyset	{HighBP(b)}	\emptyset	0.018
w_5	{hasBP(p, b), HighBP(b)}	\emptyset	{HighBP(b)}	\emptyset	\emptyset	0.252
w_6	{hasBP(p, b)}	\emptyset	{HighBP(b)}	\emptyset	\emptyset	0.108
w_7	{hasBP(p, b), HighBP(b)}	\emptyset	\emptyset	\emptyset	\emptyset	0.028
w_8	{hasBP(p, b)}	\emptyset	\emptyset	\emptyset	\emptyset	0.012

Table 1: Probability space of example TPKB.

An \mathcal{EL} TPKB is now a tuple $\langle \mathcal{T}, (\mathcal{A}_i)_{i \in \llbracket 1, n \rrbracket} \rangle$, where \mathcal{T} is an \mathcal{EL} TBox and $(\mathcal{A}_i)_{i \in \llbracket 1, n \rrbracket}$ is a sequence of n ipABoxes. Given a TPKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{i \in \llbracket 1, n \rrbracket} \rangle$, the set $\Omega_{\mathcal{K}}$ of *possible worlds of* \mathcal{K} contains all sequences $w = (\mathcal{A}'_i)_{i \in \llbracket 1, n \rrbracket}$ of classical ABoxes such that for every $i \in \llbracket 1, n \rrbracket$ and $\alpha \in \mathcal{A}'_i$, \mathcal{A}_i contains an axiom of the form $\alpha : p$. Each TPKB uniquely defines a probability space $\langle \Omega_{\mathcal{K}}, \mu_{\mathcal{K}} \rangle$, where $\mu_{\mathcal{K}} : 2^{\Omega_{\mathcal{K}}} \rightarrow [0, 1]$ satisfies

$$\mu_{\mathcal{K}}(\{(\mathcal{A}'_i)_{i \in \llbracket 1, n \rrbracket}\}) = \prod_{\substack{i \in \llbracket 1, n \rrbracket \\ \alpha : p \in \mathcal{A}_i \\ \alpha \in \mathcal{A}'_i}} p \cdot \prod_{\substack{i \in \llbracket 1, n \rrbracket \\ \alpha : p \in \mathcal{A}_i \\ \alpha \notin \mathcal{A}'_i}} (1 - p)$$

and for $W \subseteq \Omega_{\mathcal{K}}$, $\mu_{\mathcal{K}}(W) = \sum_{w \in W} \mu(\{w\})$. Intuitively, $\mu_{\mathcal{K}}(W)$ gives the probability of being in one of the possible worlds in W , by summing up the probabilities of each possible world. The definition of $\mu_{\mathcal{K}}(W)$ reflects the assumption that all probabilities in the TPKB are statistically independent.

Example 1. We define the TPKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{i \in \llbracket 1, 5 \rrbracket} \rangle$ where \mathcal{T} contains the GCI

$$\text{HighBloodPressurePatient} \equiv \exists \text{hasBloodPressure.HighBloodPressure}$$

and the ABoxes $\mathcal{A}_1 = \{\text{hasBP}(p, b), \text{HighBP}(b) : 0.7\}$, $\mathcal{A}_2 = \emptyset$, $\mathcal{A}_3 = \{\text{HighBP}(b) : 0.9\}$, $\mathcal{A}_4 = \{\text{HighBP}(b) : 0.6\}$ and $\mathcal{A}_5 = \emptyset$, where BP is short for BloodPressure. Every possible world $w = (\mathcal{A}'_i)_{i \in \llbracket 1, 5 \rrbracket}$ with $\text{hasBP}(p, b) \notin \mathcal{A}'_1$ has probability $\mu_{\mathcal{K}}(w) = 0$. The remaining possible worlds are shown in Figure 1, with the probability measure $\mu_{\mathcal{K}}$ shown in the last column.

A *model* of a TPKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{i \in \llbracket 1, n \rrbracket} \rangle$ is a mapping ι from possible worlds $w = (\mathcal{A}'_i)_{i \in \llbracket 1, n \rrbracket} \in \Omega_{\mathcal{K}}$ to sequences $(\iota(w)_i)_{i > 0}$ of (classical) models of \mathcal{T} s.t. for all $i \in \llbracket 1, n \rrbracket$, $\iota(w)_i$ is a model of the classical knowledge base $\langle \mathcal{T}, \mathcal{A}'_i \rangle$, and all $\iota(w)_i$ have the same set Δ^ι of domain elements (constant domain assumption).

Rigid Names. As typical for temporal knowledge bases, we may assume in addition a set \mathbf{N}_{rig} of *rigid names*, containing the set $\mathbf{N}_{\text{Crig}} \subseteq \mathbf{N}_{\text{C}}$ of *rigid concept names* and the set $\mathbf{N}_{\text{Rrig}} \subseteq \mathbf{N}_{\text{R}}$ of *rigid role names*. Rigid names denote names whose interpretation is independent of the flow of time. We say that a model ι of a TPKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{i \in \llbracket 1, n \rrbracket} \rangle$ *respects rigid names* iff for all $w \in \Omega_{\mathcal{K}}$, $i, j \in \llbracket 1, n \rrbracket$ and $X \in \mathbf{N}_{\text{rig}}$, $X^{\iota(w)_i} = X^{\iota(w)_j}$. Allowing for rigid names often has a direct impact on complexity and decidability of common reasoning problems, which is why typically different cases based on whether $\mathbf{N}_{\text{Crig}} = \emptyset$ or $\mathbf{N}_{\text{Rrig}} = \emptyset$ are studied for complexity.

Example 2. In the above example, the relation `hasBP` is rigid, as its interpretation should be independent of time, while the concept `HighBP` is not rigid, as the blood pressure of a patient can change from high to not high. As a consequence, the individual p will be related to the blood pressure b at all time points, even though the assertion `hasBP`(p, b) is only placed in the ipABox \mathcal{A}_1 .

ϕ	$\iota, w, i \models \phi$ iff	ϕ	$\iota, w, i \models \phi$ iff
$\exists \vec{y}.\psi(\vec{y})$	$\iota(w), i \models \exists \vec{y}.\psi(\vec{y})$	$\neg\phi$	$\iota, w, i \not\models \phi$
$\phi_1 \wedge \phi_2$	$\iota, w, i \models \phi_1$ and $\iota, w, i \models \phi_2$	$\phi_1 \vee \phi_2$	$\iota, w, i \models \phi_1$ or $\iota, w, i \models \phi_2$
$\bigcirc\phi_1$	$\iota, w, i + 1 \models \phi_1$	$\diamond\phi_1$	$\iota, w, j \models \phi_1$ for some $j \geq i$
$\square\phi_1$	$\iota, w, j \models \phi_1$ for all $j \geq i$	$\phi_1 \mathcal{U}\phi_2$	$\iota, w, j \models \phi_2$ for some $j \geq i$ and $\iota, w, k \models \phi_1$ for all $k \in \llbracket i, j - 1 \rrbracket$
$\bigcirc^-\phi_1$	$\iota, w, i - 1 \models \phi_1$ and $i > 0$	$\diamond^-\phi_1$	$\iota, w, j \models \phi_1$ for some $j \leq i$
$\square^-\phi_1$	$\iota, w, j \models \phi_1$ for all $j \leq i$	$\phi_1 \mathcal{S}\phi_2$	$\iota, w, j \models \phi_2$ for some $j \leq i$ and $\iota, w, k \models \phi_1$ for all $k \in \llbracket j + 1, i \rrbracket$
$P_{\sim p}\phi$	$\mu_{\mathcal{K}}(\{w \in \Omega_{\mathcal{K}} \mid \iota, w, i \models \phi\}) \sim p$, where $\sim \in \{<, =, >\}$		

Table 2: Entailment of Boolean TPQs in the possible world w at time point i under interpretation ι .

Temporal Probabilistic Queries. A *temporal probabilistic query* (TPQ) is of one of the following forms, where q is a CQ, ϕ_1 and ϕ_2 are a TPQs and $p \in [0, 1]$.

$$q \mid \neg\phi_1 \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \bigcirc\phi_1 \mid \diamond\phi_1 \mid \square\phi_1 \mid \phi_1 \mathcal{U}\phi_2 \\ \bigcirc^-\phi_1 \mid \diamond^-\phi_1 \mid \square^-\phi_1 \mid \phi_1 \mathcal{S}\phi_2 \mid P_{>p}\phi_1 \mid P_{=p}\phi_1 \mid P_{<p}\phi_1$$

The operators \bigcirc (next), \diamond (eventually), \mathcal{U} (until) and their inverses are temporal operators of LTL, while $P_{>}$, $P_{=}$ and $P_{<}$ are the probability operators that we add to this language. TPQs without probability operators corresponds to *temporal queries* (TQs) investigated in [8]. Note that due the disjunction operator, we can also express *unions of conjunctive queries* (UCQs), which are simply disjunctions of CQs. The answer variables of a TPQ ϕ are the answer variables of the CQs occurring in ϕ . A TPQ ϕ is *Boolean* if every variable in ϕ is bound by an existential quantifier.

We define the semantics of TPQs. Note that each possible world $w \in \Omega_{\mathcal{K}}$ has its own time line, while a model of \mathcal{K} contains a sequence of models for every possible world. For a given model, we define the semantics of temporal operators with respect to a single time line, that is, with respect to a current possible world. Probabilistic expressions $P_{\sim p}\phi$ are the only expressions that are interpreted with respect to other possible worlds.

Let ι be a model of \mathcal{K} , and ϕ a Boolean TPQ. For a single possible world $w \in \Omega_{\mathcal{K}}$ and a time point i , we say that ϕ is satisfied at w, i under ι , in symbols $\iota, w, i \models \phi$ iff the conditions in Table 2 are satisfied. Note that the temporal operators refer to the time line of a single possible world, for which they are defined as in [8]. In contrast, the probabilistic operator refers to the current time point in multiple possible worlds, and is defined similar to the probabilistic concept constructor in the DL Prob- \mathcal{ALC} [21]. A Boolean TPQ ϕ is *satisfied* in an interpretation ι at i , in symbols $\iota, i \models \phi$, iff $\iota, w, i \models \phi$ for all $w \in \Omega_{\mathcal{K}}$. It is *entailed* by the TPKB \mathcal{K} at i iff $\iota, i \models \phi$ for all models ι of \mathcal{K} . ϕ is *satisfiable in \mathcal{K} at i* iff there exists a model ι of \mathcal{K} s.t. $\iota, i \models \phi$.

Now given a TPKB \mathcal{K} , a TPQ ϕ with answer variables \vec{x} , a time point $i > 0$, and a mapping $\sigma : \vec{x} \rightarrow \mathbb{N}_1$, σ is a *certain answer for ϕ in \mathcal{K} at i* iff $\mathcal{K}, i \models \phi'$, where ϕ' is the result of applying σ on ϕ . As common, since computing answers for TPQs can be seen as a search problem that uses Boolean TPQ entailment, we focus on the decision problem of query entailment, and may refer to Boolean TPQs simply as TPQs.

0	1	2	3	→	0	1	2	3	→
1	2	3	→	0	1	2	3	→	0
2	3	→	0	1	2	3	→	0	1
3	→	0	1	2	3	→	0	1	2

...

Figure 1: Illustration of the tilings represented by the possible worlds.

Example 3. If we consider a slight variation of the query from the introduction.

$$P_{>.8}(\bigcirc^{-5}\diamond(\text{HighBPPatient}(x) \wedge \bigcirc\diamond\text{HighBPPatient}(x)))$$

For $x = p$ and time point 5, the query below the probability operator is entailed in every model of the possible worlds w_1, w_2, w_3 and w_5 , which together have a probability of 0.834. Consequently, b is an answer to the query at time point 5. Now consider the variation where the probability operators are moved inside:

$$\bigcirc^{-5}\diamond(P_{>.8}(\text{HighBPPatient}(x)) \wedge \bigcirc\diamond P_{>.8}(\text{HighBPPatient}(x)))$$

This corresponds to the situation where at least twice in the last 5 minutes, the probability of having a high blood pressure was above 0.8. As this probability is only once above this bound, this query is not entailed.

4 Lower Complexity Bound

Temporal query answering without probabilities is PSPACE-complete in combined complexity if $N_{\text{Rig}} = \emptyset$, and otherwise CONEXPTIME-complete [8]. On the other hand, computing the probability of a CQ from an ipABox is PP^{NP} -complete (see appendix), and thus also in PSPACE [30]. It turns out that, if both the temporal and the probabilistic dimension are combined, we obtain an increase to EXPSpace in complexity. This complexity increase already happens without any rigid symbols, and for TPKBs without TBox and with only one ABox, so that the DL is in fact irrelevant for this result.

A query ϕ is entailed by a TPKB \mathcal{K} iff $\neg\phi$ is not satisfiable in \mathcal{K} . As the complexity class EXPSpace is closed under complement, we can therefore focus on the problem of query satisfiability. We obtain EXPSpace-hardness by reduction of the exponential variant of the corridor tiling problem [18]. In this problem, we are given a set T of *tile types*, two special tile types $t_s, t_e \in T$, a natural number n , and two functions v and h of *compatibility constraints* $v : T \rightarrow 2^T$ (vertical) and $h : T \rightarrow 2^T$ (horizontal). The input is an *instance* of the exponential corridor tiling problem if there exists a number $m \in \mathbb{N}$ and a *tiling* $f : \llbracket 0, m \rrbracket \times \llbracket 0, 2^n - 1 \rrbracket \rightarrow T$ such that $f(0, 0) = t_s$, $f(m, 0) = t_e$, and for all $x \in \llbracket 0, m \rrbracket$ and $y \in \llbracket 0, 2^n - 1 \rrbracket$, if $x < m$, $f(x + 1, y) \in h(f(x, y))$ and if $y < 2^n - 1$, $f(x, y + 1) \in v(f(x, y))$.

We only sketch the idea of the construction here, and leave the details to the long version of the paper. We use n concept names A_i to mark the different possible worlds $w \in \Omega_{\mathcal{K}}$ with a counter, such that in interpretations ι that satisfy both the TPQ and the TPKB, $\iota, w, j \models A_i(a)$ iff the i th bit of the counter is 1 at time point j , and $\iota, w, j \not\models A_i(a)$ iff the i th bit is 0 at time point j . Furthermore, we make sure that each possible world is at each time point uniquely determined by its counter value. For this, we use the ipABox $\mathcal{A}_1 = \{A_i(a) \sim 0.5 \mid i \in \llbracket 1, n \rrbracket\}$. The query then makes sure that the counter values are increased for each time point. Figure 1 illustrates this idea. Intuitively, each possible world corresponds to a row in the tiling, with its counter value at time point 1 denoting the row number.

At each time point, there are two possible worlds that can be most easily recognised by a query: the one whose counter value is 0 (which satisfies the query $\bigwedge_{1 \leq i \leq n} \neg A_i(a)$), and the one whose counter value is $2^n - 1$ (which satisfies the query $\bigwedge_{1 \leq i \leq n} A_i(a)$). Unless the latter one represents the last row, both these possible worlds correspond to neighbouring rows, which means at each time point we can recognise the vertical neighbour relation for two rows easily, and thus enforce tiling conditions in that direction with the following query, where $L(a)$ is an assertion that marks the last row, and for a tile type $t \in T$, $B_t(a)$ expresses that the current cell has a tile of type t .

$$\square \bigwedge_{t_1 \in T} \left(\left(B_{t_1}(a) \wedge \bigwedge_{i \in [1, n]} A_i(a) \wedge \neg L(a) \right) \rightarrow \bigvee_{t_2 \in v(t_1)} \text{P}_{=1} \left(\left(\bigwedge_{i \in [1, n]} \neg A_i(a) \right) \rightarrow B_{t_2}(a) \right) \right)$$

As we can only check the vertical tiling conditions for one pair of rows at a time, we represent each cell by up to 2^n succeeding time points in each possible worlds, performing a switch only when the counter reaches $2^n - 1$. The remaining reduction is described in the appendix.

Lemma 4. *Entailment of TPQs is EXPSpace-hard in combined complexity, even for TKBs $\langle \mathcal{T}, (\mathcal{A}_i)_{i \in [1, n]} \rangle$ where $\mathcal{T} = \emptyset$, $n = 1$ and $\mathbf{N}_{\text{Crig}} = \mathbf{N}_{\text{Rrig}} = \emptyset$.*

5 Upper Complexity Bound

We show that the complexity result presented in the last section are indeed tight, even if $\mathbf{N}_{\text{Rrig}} \neq \emptyset$. We sketch here only the case without rigid symbols. How rigid symbols are integrated is then discussed in the appendix. Our construction is based on an abstraction of a temporal probabilistic model, which we call quasimodel, which collects for each time point and possible world the CQs occurring in the input query that are entailed, as well as the CQs that are not entailed. We focus on satisfiability of a TPQ ϕ in a TPKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{i \in [1, n]} \rangle$, where we say ϕ is satisfiable in \mathcal{K} iff ϕ is satisfiable in \mathcal{K} at 1. In other words, we ignore the time point to make things simpler. Since ϕ is satisfiable in \mathcal{K} at i iff $\circ^{i-1}\phi$ is satisfiable, this is sufficient for our complexity analysis.

We can assume without loss of generality that ϕ contains only the operators $\wedge, \neg, \mathcal{U}, \mathcal{S}$ and $\text{P}_{\sim p}$, since the remaining operators can be linearly encoded using known equivalences. Denote by $\text{sub}(\phi)$ the sub-queries of ϕ and set $T(\phi) = \{\psi, \neg\psi \mid \psi \in \text{sub}(\phi)\}$. A *quasi-state* is a mapping $Q : \Omega_{\mathcal{K}} \rightarrow T(\phi)$ that satisfies the following conditions:

- S1** $\neg\psi \in Q_i(w)$ iff $\psi \notin Q_i(w)$,
- S2** for all $\psi_1 \wedge \psi_2 \in T(\phi)$: $\psi_1 \wedge \psi_2 \in Q_i(w)$ iff $\psi_1 \in Q_i(w)$ and $\psi_2 \in Q_i(w)$, and
- S3** for all $\text{P}_{\sim p}(\psi) \in T(\phi)$: $\text{P}_{\sim p}(\psi) \in Q_i(w)$ iff $\mu_{\mathcal{K}}(\{w \mid \psi \in Q_i(w)\}) \sim p$.

The quasistate abstracts probabilistic interpretations at a single time point by assigning queries to each possible world according to the semantics of the atemporal operators in our query language. To incorporate the temporal dimension, we consider unbounded sequences of quasistates $(Q_i)_{i \geq 1}$, which we call *quasimodels for ϕ in \mathcal{K}* , and which have to satisfy the following conditions for $i \geq 1$ and $w = (A'_i)_{i \in [1, n]} \in \Omega_{\mathcal{K}}$.

- Q1** $\phi \in Q_1(w)$,

- Q2** if $i \in \llbracket 1, n \rrbracket$, $\mathcal{A}'_i \models \bigwedge_{\psi \in X} \psi$, where $X = \{\psi \in Q_i(w) \mid \psi \text{ is a CQ or a negated CQ}\}$.
- Q3** for all $\circ\psi \in T(\phi)$, $\circ\psi \in Q_i(w)$ iff $\psi \in Q_{i+1}(w)$,
- Q4** for all $\circ^-\psi \in T(\phi)$, $\circ^-\psi \in Q_{i+1}(w)$ iff $\psi \in Q_i(w)$,
- Q5** for all $\psi_1 \mathcal{U} \psi_2 \in T(\phi)$, $\psi_1 \mathcal{U} \psi_2 \in Q_i$ iff there exists $j \geq i$ s.t. $\psi_2 \in Q_j(w)$ and for all $k \in \llbracket i, j-1 \rrbracket$, $\psi_1 \in Q_k(w)$, and
- Q6** for all $\psi_1 \mathcal{S} \psi_2 \in T(\phi)$, $\psi_1 \mathcal{S} \psi_2 \in Q_i$ iff there exists $j \leq i$ s.t. $\psi_2 \in Q_j(w)$ and for all $k \in \llbracket j-1, i \rrbracket$, $\psi_1 \in Q_k(w)$.

Again, the intuition behind these conditions follows directly from the semantics of the temporal operators. As we show in the appendix, quasimodels are indeed sufficient to witness the satisfiability of a TPQ in a TPKB. Moreover, it is sufficient to consider quasi-models that are of a certain regular form, which is the crucial element for our complexity bound.

Lemma 5. *ϕ is satisfiable in \mathcal{K} with $N_{\text{Crig}} = N_{\text{Rrig}} = \emptyset$ iff there exists a quasi-model for ϕ in \mathcal{K} wrt. \mathcal{S} and \mathcal{U} which is of the form*

$$Q_1, \dots, Q_m (Q_{m+1}, \dots, Q_{m+o})^\omega,$$

where m and o are both double-exponentially bounded in the size of \mathcal{K} and ϕ .

Exploiting the fact that $\text{EXPSPACE} = \text{NEXPSPACE}$, we obtain our space bounds by a non-deterministic decision procedure that can be roughly sketched as follows. We first guess the numbers m and o . While m and o are double-exponentially bounded, they can be stored in exponential space using binary encoding. We now guess the quasistates Q_1, \dots, Q_{m+o} one after the other, where we carefully make sure that all conditions of quasimodels are satisfied. In particular, we keep track of \mathcal{U} - and \mathcal{S} -formulae that have to be satisfied, and we keep the state Q_{m+1} in memory to test that it is compatible to Q_{m+o} , and that all \mathcal{U} -formulae in Q_{m+1} are satisfied before we reach Q_{m+o} .

In appendix, we present a refined version of quasimodels, which also have the above regularity property, but additionally take into consideration rigid predicates. The main idea is to use for each possible world an additional structure that determines which sets of CQs and their negations can be entailed at any time point under the rigidity constraints. This structure takes exponential space per possible world, and can be computed in non-deterministic exponential time.

Theorem 6. *Entailment of TPQs from DL-Lite/ \mathcal{EL} -TPKBs can be decided in EXPSPACE, even if $N_{\text{Rrig}} \neq \emptyset$ and $N_{\text{Crig}} \neq \emptyset$.*

6 Removing Negation

The complexity increase discussed in the last sections can be avoided if we restrict ourselves to *positive TPQs*, which are TPQs that do not use the operators \neg , $P_{<p}$ and $P_{=p}$. Note that the probability operators $P_{<p}\phi$ and $P_{=p}\phi$ can be seen as implicit negation operators, as they express the non-entailment of ϕ in some possible worlds, whereas $P_{>p}\phi$ only expresses the positive entailment of ϕ in possible worlds. The example queries shown in this paper are all positive queries.

In the absence of negation, it is possible to evaluate the probabilities of sub-queries “inside-out”, starting from queries of the form $P_{>p}\phi$ where ϕ contains no probabilistic operators. For

non-probabilistic temporal queries, it can be decided in P data and NP combined complexity whether they are entailed. This allows to decide the entailment of $P_{>p}\phi$ at any time point in PP, by using a probabilistic Turing machine that guesses all possible worlds of the TPKB. Using closure properties of the complexity class PP, we can thus obtain tight complexity bounds for the case where the nesting depth of probability operators is bounded, and otherwise inclusion in P^{PP} , a complexity class that is still contained in PSPACE.

Theorem 7. *Entailment of positive TPQs from \mathcal{EL} -TPKBs is PP-complete wrt. data complexity. Regarding combined complexity, it is PP^{NP} -complete if the nesting depth of probability-operators in the query is bounded, and otherwise in P^{PP} . The results already hold for $N_{Rig} \neq \emptyset$.*

7 Conclusion

We investigated the complexity of querying temporal probabilistic data using a combination of LTL and conjunctive queries with probability operators. While pure temporal and pure probabilistic query answering are both in PSPACE for most cases, combining both dimensions yields completeness for EXPSpace. This increase in complexity already happens without TBoxes and just with a single ABox, so that the hardness result is in fact independent of DL reasoning. This increase of complexity can be avoided if we restrict ourselves to positive TPQs, in which case the temporal dimension comes at no cost or almost no cost compared to pure probabilistic query answering. While this paper presented a theoretical study of the setting of temporal probabilistic query answering, the methods presented give no clear idea how a practical implementation would look like. For description logics that enjoy first-order rewritability such as *DL-Lite*, a solution could be to rewrite temporal queries into SQL and use a probabilistic database system to compute their probabilities. However, this approach would only work for queries that do not use negation, and it is not clear whether it can be used with rigid symbols [8]. Another open question is how the data complexity looks like in the case where we allow for negation, and whether the complexities further change if we admit more expressive DLs, or even DLs that support temporal and probabilistic operators themselves.

References

- [1] Artale, A., Kontchakov, R., Kovtunova, A., Ryzhikov, V., Wolter, F., Zakharyashev, M.: Ontology-mediated query answering over temporal data: A survey (invited talk)
- [2] Baader, F., Borgwardt, S., Koopmann, P., Ozaki, A., Thost, V.: Metric temporal description logics with interval-rigid names. In: Proceedings of FroCoS 2017. pp. 60–76 (2017). https://doi.org/10.1007/978-3-319-66167-4_4
- [3] Baader, F., Borgwardt, S., Lippmann, M.: Temporal query entailment in the description logic *SHQ*. J. Web Sem. **33**, 71–93 (2015). <https://doi.org/10.1016/j.websem.2014.11.008>
- [4] Baader, F., Ghilardi, S., Lutz, C.: LTL over description logic axioms. ACM Trans. Comput. Log. **13**(3), 21:1–21:32 (2012). <https://doi.org/10.1145/2287718.2287721>
- [5] Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An Introduction to Description Logic. Cambridge University Press (2017)
- [6] Baader, F., Koopmann, P., Turhan, A.: Using ontologies to query probabilistic numerical data. In: Proceedings of FroCoS 2017. pp. 77–94 (2017). https://doi.org/10.1007/978-3-319-66167-4_5

- [7] Borgwardt, S., Ceylan, İ.İ., Lukasiewicz, T.: Ontology-mediated queries for probabilistic databases. In: Proceedings of AAAI 2017. pp. 1063–1069 (2017), <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14365>
- [8] Borgwardt, S., Lippmann, M., Thost, V.: Temporalizing rewritable query languages over knowledge bases. *J. Web Sem.* **33**, 50–70 (2015). <https://doi.org/10.1016/j.websem.2014.11.007>
- [9] Borgwardt, S., Thost, V.: Temporal query answering in dl-lite with negation. In: Proceedings of GCAI 2015. pp. 51–65 (2015), <http://www.easychair.org/publications/paper/245305>
- [10] Borgwardt, S., Thost, V.: Temporal query answering in the description logic \mathcal{EL} . In: Proceedings of IJCAI 2015. pp. 2819–2825 (2015), <http://ijcai.org/Abstract/15/399>
- [11] Bourgaux, C., Turhan, A.: Temporal query answering in DL-Lite over inconsistent data. In: The Semantic Web - ISWC 2017. pp. 121–137 (2017). https://doi.org/10.1007/978-3-319-68288-4_8
- [12] Brandt, S., Kalayci, E.G., Kontchakov, R., Ryzhikov, V., Xiao, G., Zakharyashev, M.: Ontology-based data access with a Horn fragment of metric temporal logic. In: Proceedings of AAAI 2017. pp. 1070–1076 (2017), <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14881>
- [13] Ceylan, İ.İ., Darwiche, A., den Broeck, G.V.: Open-world probabilistic databases: An abridged report. In: Proceedings of IJCAI 2017. pp. 4796–4800 (2017). <https://doi.org/10.24963/ijcai.2017/669>
- [14] Ceylan, İ.İ., Peñaloza, R.: Dynamic Bayesian description logics. In: Proceedings of DL 2015 (2015), <http://ceur-ws.org/Vol-1350/paper-48.pdf>
- [15] Ceylan, İ.İ., Peñaloza, R.: The Bayesian ontology language \mathcal{BEL} . *J. Autom. Reasoning* **58**(1), 67–95 (2017). <https://doi.org/10.1007/s10817-016-9386-0>
- [16] Dalvi, N.N., Schnaitter, K., Suciu, D.: Computing query probability with incidence algebras. In: Proceedings of PODS 2010. pp. 203–214 (2010). <https://doi.org/10.1145/1807085.1807113>
- [17] Elkin, P.L., Brown, S.H., Husser, C.S., Bauer, B.A., Wahner-Roedler, D., Rosenbloom, S.T., Speroff, T.: Evaluation of the content coverage of SNOMED CT: ability of SNOMED clinical terms to represent clinical problem lists. *Mayo Clin. Proc.* **81**(6), 741–748 (2006)
- [18] van Emde Boas, P.: The convenience of tilings. *Lecture Notes in Pure and Applied Mathematics* pp. 331–363 (1997)
- [19] Fortnow, L., Reingold, N.: PP is closed under truth-table reductions. In: Proceedings of the Sixth Annual Structure in Complexity Theory Conference. pp. 13–15 (1991). <https://doi.org/10.1109/SCT.1991.160239>
- [20] Gabbay, D.M., Kurucz, A., Wolter, F., Zakharyashev, M.: Many-dimensional modal logics: Theory and applications. Elsevier (2003)
- [21] Gutiérrez-Basulto, V., Jung, J.C., Lutz, C., Schröder, L.: Probabilistic description logics for subjective uncertainty. *J. Artif. Intell. Res.* **58**, 1–66 (2017). <https://doi.org/10.1613/jair.5222>
- [22] Gutiérrez-Basulto, V., Jung, J.C., Ozaki, A.: On metric temporal description logics. In: Proceedings of ECAI 2016. pp. 837–845 (2016). <https://doi.org/10.3233/978-1-61499-672-9-837>

- [23] Jung, J.C., Lutz, C.: Ontology-based access to probabilistic data with OWL QL. In: The Semantic Web - ISWC 2012. Lecture Notes in Computer Science, vol. 7649, pp. 182–197. Springer (2012). https://doi.org/10.1007/978-3-642-35176-1_12
- [24] Kumar, N., Khunger, M., Gupta, A., Garg, N.: A content analysis of smartphone-based applications for hypertension management. *Journal of the American Society of Hypertension* **9**(2), 130–136 (2015)
- [25] Lukasiewicz, T.: Expressive probabilistic description logics. *Artif. Intell.* **172**(6-7), 852–883 (2008). <https://doi.org/10.1016/j.artint.2007.10.017>
- [26] Lutz, C., Wolter, F., Zakharyashev, M.: Temporal description logics: A survey. In: Proceedings of TIME 2008. pp. 3–14. IEEE Press (2008). <https://doi.org/10.1109/TIME.2008.14>
- [27] Margara, A., Urbani, J., van Harmelen, F., Bal, H.: Streaming the web: Reasoning over dynamic data. *Web Semantics: Science, Services and Agents on the World Wide Web* **25**, 24 – 44 (2014). <https://doi.org/https://doi.org/10.1016/j.websem.2014.02.001>
- [28] Rector, A., Gangemi, A., Galeazzi, E., Glowinski, A., Rossi-Mori, A.: The GALEN CORE model schemata for anatomy: Towards a re-usable application-independent model of medical concepts. In: Proc. MIE’94. pp. 229–233 (1994)
- [29] Ronca, A., Kaminski, M., Grau, B.C., Motik, B., Horrocks, I.: Stream reasoning in temporal datalog. *CoRR* **abs/1711.04013** (2017), <http://arxiv.org/abs/1711.04013>
- [30] Toda, S.: PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.* **20**(5), 865–877 (1991). <https://doi.org/10.1137/0220053>
- [31] Wagner, K.W.: The complexity of combinatorial problems with succinct input representation. *Acta Informatica* **23**(3), 325–356 (Jun 1986). <https://doi.org/10.1007/BF00289117>

A Appendix

A.1 Lower Bounds

Lemma 4. *Entailment of TPQs is EXPSpace-hard in combined complexity, even for TKBs $\langle \mathcal{T}, (\mathcal{A}_i)_{i \in [1, n]} \rangle$ where $\mathcal{T} = \emptyset$, $n = 1$ and $\mathbf{N}_{\text{Crig}} = \mathbf{N}_{\text{Rrig}} = \emptyset$.*

Proof. We provide a reduction of the EXPSpace-complete 2^n corridor tiling problem as specified in the main text. We provide an encoding of this problem using a single ipABox \mathcal{A} and a TPCQ ϕ with negations. \mathcal{A} and ϕ are constructed in such a way that there is a correspondence between solutions f to the tiling problem and models ι of $\mathcal{K} = \langle \emptyset, (\mathcal{A}) \rangle$ and ϕ . The (bounded) vertical dimension of the corridor is represented across the 2^n possible worlds, while the (unbounded) horizontal dimension is represented along the time line. Specifically, the correspondence from ι to a tiling f is specified via mappings $y : [0, 2^n - 1] \rightarrow \Omega_{\mathcal{K}}$ and $c : \mathbb{N} \times \Omega_{\mathcal{K}} \rightarrow T$ s.t. the tiling is provided by $f(i, j) = c(2^n \cdot j + i, y(j))$.

We use a concept name B_t for every tile type $t \in T$, and use the assertion $B_t(a)$ to denote that the cell corresponding to a possible world/time point pair has a tile of type t . To make sure that every pair of a possible world and a time point represents exactly one $t \in T$, we use the query

$$\square \bigwedge_{B_{t_1}(a) \in T} \left(t_1 \leftrightarrow \bigwedge_{t_2 \in T, t_1 \neq t_2} \neg B_{t_2}(a) \right).$$

We can already define the first mapping c from pairs of possible worlds and time points to their tile type: given a model ι of the final query, we define $c(i, w) = t$, where $\iota, w, i \models B_t(a)$.

To provide for the mapping $y : [0, 2^n - 1] \rightarrow \Omega_{\mathcal{K}}$, which assigns row numbers to possible worlds, we set

$$\mathcal{A}_1 = \{A_i(a) : 0.5 \mid i \in \llbracket 1, n \rrbracket\},$$

where A_1, \dots, A_n are concept names that correspond to bits in a binary counter. Since all probabilities are statistically independent, each counter value is represented by some possible world. For $i \in [0, 2^n - 1]$, the value of $y(i)$ is then simply the possible world in which the counter has the value i at the first time point. We mark the possible world which represents the last row with the assertion $L(a)$ and the following queries.

$$\begin{aligned} & \left(\bigwedge_{i \in \llbracket 1, n \rrbracket} A_i(a) \right) \leftrightarrow L(a) \\ & \square (L(a) \leftrightarrow \bigcirc L(a)) \end{aligned}$$

The previous queries ensure that the mappings y and c are well-defined. The following query ensures the tiling conditions regarding the special tiles t_s and t_e , which have to occur in the first row of respectively the first and the last column of the tiling solution.

$$\left(\bigwedge_{i \in \llbracket 1, n \rrbracket} \neg A_i(a) \right) \rightarrow (B_{t_s}(a) \wedge \diamond B_{t_e}(a))$$

It remains to provide queries that enforce the compatibility constraints.

The following queries ensure that the counters in each possible world get incremented across the time line.

$$\begin{aligned} & \square \bigwedge_{i \in \llbracket 1, n \rrbracket} \left(\left(\neg A_i(a) \wedge \bigwedge_{j < i} A_j(a) \right) \rightarrow \bigcirc \left(A_i(a) \wedge \bigwedge_{j < i} \neg A_j(a) \right) \right) \\ & \square \bigwedge_{i \in \llbracket 1, n \rrbracket} \left(\left(\neg A_i(a) \wedge \bigvee_{j < i} \neg A_j(a) \right) \rightarrow \bigcirc \neg A_i(a) \right) \\ & \square \bigwedge_{i \in \llbracket 1, n \rrbracket} \left(\left(A_i(a) \wedge \bigvee_{j < i} \neg A_j(a) \right) \rightarrow \bigcirc A_i(a) \right) \\ & \square \left(\left(\bigwedge_{i \in \llbracket 1, n \rrbracket} A_i(a) \right) \rightarrow \bigcirc \bigwedge_{i \in \llbracket 1, n \rrbracket} \neg A_i(a) \right) \end{aligned}$$

In each possible world, the current tile type is transported to the next time point until until the counter reaches $2^n - 1$.

$$\square \bigwedge_{t \in T} \left(\left(B_t(a) \wedge \bigvee_{i \in \llbracket 1, n \rrbracket} \neg A_i(a) \right) \rightarrow \bigcirc B_t(a) \right)$$

If the counter in a possible world reaches $2^n - 1$, we can identify the world that corresponds to the next row easily, as its counter then has the value 0. We can thus enforce the vertical

compatibility constraints using the following query.

$$\begin{aligned} & \square \bigwedge_{t_1 \in T} \left(\left(B_{t_1}(a) \wedge \neg L(a) \wedge \bigwedge_{i \in \llbracket 1, n \rrbracket} A_i(a) \right) \right. \\ & \quad \left. \rightarrow \bigvee_{t_2 \in v(t_1)} \text{P}_{=1} \left(\left(\bigwedge_{i \in \llbracket 1, n \rrbracket} \neg A_i(a) \right) \rightarrow B_{t_2}(a) \right) \right) \end{aligned}$$

To enforce the horizontal constraints, we only have to identify the next time point when the counter is $2^n - 1$.

$$\square \bigwedge_{t_1 \in T} \left(\left(B_{t_1}(a) \wedge \bigwedge_{i \in \llbracket 1, n \rrbracket} A_i(a) \right) \rightarrow \bigcirc \bigvee_{t_2 \in h(t_1)} B_{t_2}(a) \right)$$

The final query ϕ is the conjunction of all queries. It is now standard to verify that the tiling problem has a solution iff ϕ is satisfiable in \mathcal{A} . \square \square

A.2 Upper Bound

We first extend the quasi-models introduced in the main text to take into account rigid names. Specifically, we have to make sure that the quasimodel corresponds to an interpretation ι such that in every possible world w and for all $i, j \geq 1$ and $X \in \mathbf{N}_{\text{rig}}$, $X^{\iota(w)_i} = X^{\iota(w)_j}$. Note that our quasimodels abstract real interpretation by only considering the queries that are entailed at each time point. Let $\{q_1, \dots, q_n\}$ be the CQs that occur in the query ϕ . To specify which combinations of queries can be satisfied at any time point in the same temporal model $(\iota(w)_i)_{i \geq 1}$, we use guess a set $\mathcal{S}(w) \subseteq 2^{\{q_1, \dots, q_n\}}$ for each possible world $w \in \Omega_{\mathcal{K}}$. In addition, we use a mapping $a : \Omega_{\mathcal{K}} \times \llbracket 1, n \rrbracket \rightarrow 2^{\{q_1, \dots, q_n\}}$ to assign elements from $\mathcal{S}(w)$ to the ABoxes in the possible worlds. The following definition captures when such a set \mathcal{S} and a mapping a correspond to valid models of \mathcal{K} .

Definition 8. Given a possible world $w = (A'_i)_{i \in \llbracket 1, n \rrbracket} \in \Omega_{\mathcal{K}}$, a set $\mathcal{S}(w) = \{X_1, \dots, X_k\} \subseteq 2^{\{q_1, \dots, q_m\}}$ and a mapping $a : \Omega_{\mathcal{K}} \times \llbracket 1, n \rrbracket \rightarrow \mathcal{S}$, \mathcal{S} is called *r-satisfiable wrt. w and a* iff there exist (classical) interpretations $\mathcal{J}_1, \dots, \mathcal{J}_k, \mathcal{I}_1, \dots, \mathcal{I}_n$ such that

- E1** for any two interpretations $\mathcal{I}', \mathcal{I}'' \in \{\mathcal{J}_1, \dots, \mathcal{J}_k, \mathcal{I}_1, \dots, \mathcal{I}_n\}$, we have $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}''}$ and $X^{\mathcal{I}'} = X^{\mathcal{I}''}$ for all $X \in \mathbf{N}_{\text{rig}}$,
- E2** the interpretations are models of \mathcal{T} ,
- E3** for all $X \in \mathcal{S}$, $\mathcal{J}_i \models \bigwedge_{q \in X_i} q \wedge \bigwedge_{q \notin X_i} \neg q$, and
- E4** and for all $i \in \llbracket 1, n \rrbracket$, $\mathcal{I}_i \models \bigwedge_{q \in a(w, i)} q \wedge \bigwedge_{q \notin a(w, i)} \neg q$ and $\mathcal{I}_i \models A'_i$.

r-satisfiability provides for a sufficient abstraction of a model of a possible world, based on which satisfiability of the query can be decided.

The following is a direct consequence of the proofs for [10, Lemma 4.17] and [9, Theorem 5.1].

Lemma 9. *For a given $\mathcal{S}(w) \subseteq 2^{\{q_1, \dots, q_m\}}$ and $a : \Omega_{\mathcal{K}} \times \llbracket 1, n \rrbracket \rightarrow \mathcal{S}$, it can be decided in exponential time wrt. to \mathcal{K} and ϕ .*

To use r-satisfiability for probabilistic TPKBs $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{i \in \llbracket 1, n \rrbracket} \rangle$, for each $w \in \Omega_{\mathcal{K}}$, we guess a set $\mathcal{S}(w) \subseteq 2^{\{q_1, \dots, q_m\}}$ and a mapping $a(w) : \llbracket 1, n \rrbracket \rightarrow \mathcal{S}(w)$ s.t. $\mathcal{S}(w)$ is r-satisfiable wrt. $a(w)$ and $\langle \mathcal{T}, w \rangle$. Using \mathcal{S} and a , we specify the following additional properties on quasimodels, and speak of *quasi-models compatible to \mathcal{S} and a* .

Q7 $Q_i(w) \cap \{q_1, \dots, q_m\} \in \mathcal{S}(w)$,

Q8 if $i \in \llbracket 1, n \rrbracket$, $Q_i(w) \cap \{q_1, \dots, q_m\} = a(w, i)$.

We first show that with this extended notion, quasimodels indeed correspond to models of \mathcal{K} that satisfy ϕ and respect rigidity constraints.

Lemma 10. *ϕ is satisfiable in \mathcal{K} iff there exist mappings $\mathcal{S} : \Omega_{\mathcal{K}} \rightarrow 2^{\{q_1, \dots, q_m\}}$ and $a : \Omega_{\mathcal{K}} \times \llbracket 1, n \rrbracket \rightarrow 2^{\{q_1, \dots, q_m\}}$ s.t.*

1. for every $w \in \Omega_{\mathcal{K}}$, $\mathcal{S}(w)$ is r-satisfiable wrt. a and w , and
2. there exists a quasi-model for ϕ in \mathcal{K} wrt. \mathcal{S} and a .

Proof. (\Rightarrow) ϕ is satisfiable in \mathcal{K} . There then exists a temporal probabilistic model ι of \mathcal{K} s.t. for all $w \in \Omega_{\mathcal{K}}$, $\iota, w, 1 \models \phi$. For an interpretation $\iota(w)_i$, set $X_i(w) = \{q_i \mid \iota(w)_i \models q_i\}$. \mathcal{S} and a are now defined by setting for all $w \in \Omega_{\mathcal{K}}$ and $i \in \llbracket 1, n \rrbracket$:

$$\begin{aligned} \mathcal{S}(w) &= \{X_j(w) \mid j > 0\} \\ a(w, i) &= X_i(w). \end{aligned}$$

One easily verifies that for all $w \in \Omega_{\mathcal{K}}$, $\mathcal{S}(w)$ is r-satisfiable wrt. a and w . The quasi-model Q_1, \dots is now defined by setting

$$Q_i(w) = \{\psi \in T(\phi) \mid \iota, w, i \models \psi\}$$

for all $w \in \Omega_{\mathcal{K}}$ and $i > 1$. By checking the semantic definitions in Table 2 against Conditions **Q1–Q6** in the definition of quasi-models, it is now standard to verify that Q_1, \dots is a quasi-model for ϕ in \mathcal{K} wrt. \mathcal{S} and a .

(\Leftarrow) There exist the mappings \mathcal{S} and a , as well as a quasi model Q_1, \dots , as in the lemma. We construct a temporal probabilistic model ι of \mathcal{K} s.t. for all $w \in \Omega_{\mathcal{K}}$, $\iota, w, 1 \models \phi$. For all $w \in \Omega_{\mathcal{K}}$, assume $\mathcal{S}(w) = \{X_1^w, \dots, X_{k_w}^w\}$. By definition of r-satisfiability, for every $w \in \Omega_{\mathcal{K}}$, there exists interpretations $\mathcal{J}_1^w, \dots, \mathcal{J}_{m_w}^w, \mathcal{I}_1^w, \dots, \mathcal{I}_n^w$ as in Definition 8. ι is now defined by setting for all $w \in \Omega_{\mathcal{K}}$:

- for all $i \in \llbracket 1, n \rrbracket$: $\iota(w)_i = \mathcal{I}_i^w$, and
- for all $i > n$: $\iota(w)_i = \mathcal{J}_j^w$, where j is such that $Q_i(w) \cap \{q_1, \dots, q_m\} = X_j^w$ (by Condition **Q7**).

Clearly, ι respects rigid names. It is now standard, by checking the semantic definitions in Table 2 against Conditions **Q1–Q6**, that ι is indeed a model of \mathcal{K} and ϕ as required. $\square \square$

To obtain the EXPSPACE upper bound, it therefore suffices to show that the existence of such a quasi-model can be decided in EXPSPACE. The main insight here is that satisfiability of quasi-models can be reduced to satisfiability of certain periodic quasi-models. Note that, together with the last lemma, the following lemma is a strengthening of Lemma 5 in the main text.

Lemma 11. *There exists a quasi-model for ϕ in \mathcal{K} wrt. \mathcal{S} and a iff there exists a quasi-model for ϕ of in \mathcal{K} wrt. \mathcal{S} and a which is of the form*

$$Q_1, \dots, Q_m(Q_{m+1}, \dots, Q_{m+o})^\omega,$$

where m and o are both double-exponentially bounded in the size of \mathcal{K} and ϕ .

Proof. First note that there can be at most double-exponentially many different quasi-states in a quasi-model: $\Omega_{\mathcal{K}}$ contains at most $2^{|\mathcal{K}|}$ many elements, and for each $w \in \Omega_{\mathcal{K}}$, $Q_i(w)$ contains at most $|\phi|$ elements. We obtain that there are at most $2^{2^{|\mathcal{K}|} \cdot |\phi|}$ many different combinations. For indices i, j s.t. $Q_i = Q_j$, we define an operation *merging of Q_i and Q_j* in Q_1, \dots , which replaces the quasi-model with $Q_1, \dots, Q_i, Q_{j+1}, \dots$. One can verify that the result of merging in a quasi model is again a quasi model: 1) Conditions **Q1–Q4** only consider at most two-subsequent states, and 2) Conditions **Q6** and **Q5** are still satisfied in the new quasi-model.

Now let Q_1, \dots be any quasi-model, and let m, o be two indices s.t. $Q_m = Q_{m+o}$, and the following condition is satisfied:

(*) for every $w \in \Omega_{\mathcal{K}}$ and $\psi_1 \mathcal{U} \psi_2 \in Q_m(w)$, there exists $k < m + o$ s.t. $\psi_2 \in Q_k(w)$.

From (*), it already follows that $Q_1, \dots, Q_m(Q_{m+1}, \dots, Q_{m+o})^\omega$ is also a quasi-model. However, m and o might not be double-exponentially bounded in the size of \mathcal{K} and ϕ . By the above observation, we may assume that no quasi-state occurs twice before Q_m , since we can always merge any quasi-states that occur more than once, so that $m \leq 2^{2^{|\mathcal{K}|} \cdot |\phi|}$. To reduce the index of Q_{m+o} , we exhaustively merge any two quasi-states that occur between Q_m and Q_{m+o} for which merging does not break Condition (*). The resulting quasi-state can now be represented as

$$Q'_1, \dots, Q'_m(Q'_{m+1}, \dots, Q'_{m+o'})^\omega.$$

We give a bound on o' . For every $i, j \in \llbracket n+1, n+o' \rrbracket$ s.t. $Q'_i = Q'_j$, there must be some $w \in \Omega_{\mathcal{K}}$, $\psi_1 \mathcal{U} \psi_2 \in Q'_n(w)$, and $k \in \llbracket i, j \rrbracket$ s.t. $\psi_2 \in Q'_k(w)$ and $\psi_2 \notin Q'_l(w)$ for all $l \in \llbracket n, k-1 \rrbracket$, since otherwise Q'_i and Q'_j would have been merged. It follows that every quasi-state is repeated at most $2^{|\mathcal{K}|} \cdot |\phi|$ times, because there are at most $2^{|\mathcal{K}|}$ possible worlds in $\Omega_{\mathcal{K}}$ and for each $w \in \Omega_{\mathcal{K}}$, at most $|\phi|$ queries of the form $\psi_1 \mathcal{U} \psi_2$ in $Q'_n(w)$. Because the number of distinct quasistates is bounded by $2^{2^{|\mathcal{K}|} \cdot |\phi|}$, we obtain $o' \leq 2^{|\mathcal{K}|} \cdot |\phi| \cdot 2^{2^{|\mathcal{K}|} \cdot |\phi|}$, that is, o' is double-exponentially bounded in the size of \mathcal{K} and ϕ . It follows that we can transform any quasi-model into a quasi-model of the required form, and thus that a quasi-model exists iff there exists a regular quasi-model which is of the form as in the lemma. \square \square

We are now ready to prove the complexity upper bound.

Lemma 12. *Given the mappings $\mathcal{S} : \Omega_{\mathcal{K}} \rightarrow 2^{\{q_1, \dots, q_m\}}$ and $a : \Omega_{\mathcal{K}} \times \llbracket 1, n \rrbracket \rightarrow 2^{\{q_1, \dots, q_m\}}$, it can be decided in EXPSPACE whether there exists a quasi-model for ϕ in \mathcal{K} wrt. \mathcal{S} and a .*

Proof. By Lemma 11, there exists a quasi-model Q_1, \dots iff there exists a periodic quasi-model of the form

$$Q_1, \dots, Q_m(Q_{m+1}, \dots, Q_{m+o})^\omega,$$

where both m and o are double-exponentially bounded.

To verify the existence of such a quasi-model in (non-deterministic) exponential space, we proceed as follows. We first guess the numbers m and o , which both require at most exponentially many bits in binary representation. We then guess the quasi-states Q_i one after the other, keeping

always two preceding quasi-states in memory, and verify that they satisfy the Conditions **Q1–Q4**, and additionally that $\psi_1 \mathcal{U} \psi_2 \in Q_i(w)$ iff $\psi_2 \in Q_i(w)$ or $\psi_1 \mathcal{U} \psi_2 \in Q_{i+1}(w)$, and similarly for queries of the form $\psi_1 \mathcal{S} \psi_2$. To verify that each \mathcal{S}/\mathcal{U} -formula is eventually satisfied, we keep a set of those queries for each possible world that have not been satisfied yet, which we update at each time point. In the same manner, we check whether the negation of a \mathcal{S}/\mathcal{U} -formula is satisfied. After we guessed the quasi-state Q_{m+1} , we store this quasi-state in memory, as well as all \mathcal{S}/\mathcal{U} -queries that still have to be satisfied at this point. We then proceed until Q_{m+o+1} , and verify that all \mathcal{S}/\mathcal{U} -queries from Q_{m+1} have been satisfied in the meanwhile, and that $Q_{m+o+1} = Q_{m+1}$. Since $\text{NEXPSPACE} = \text{EXPSPACE}$, the above procedure decides existence of a quasi-model in exponential space. \square \square

Theorem 6. *Entailment of TPQs from DL-Lite/ \mathcal{EL} -TPKBs can be decided in EXPSPACE, even if $\mathbf{N}_{\text{Rrig}} \neq \emptyset$ and $\mathbf{N}_{\text{Crig}} \neq \emptyset$.*

Proof. Entailment of a query ϕ corresponds to non-satisfiability of the query $\neg\phi$, and EXPSPACE is closed under complement. To decide satisfiability of a TPQ ϕ with negations in a TPKB \mathcal{K} , we guess for each $w \in \Omega_{\mathcal{K}}$ $\mathcal{S}(w) \in 2^{\{q_1, \dots, q_m\}}$ and $a(w) : \llbracket 1, n \rrbracket \rightarrow \mathcal{S}(w)$ and verify that $\mathcal{S}(w)$ is r -satisfiable wrt $a(w)$ and $\langle \mathcal{T}, w \rangle$. By Lemma 9, this can be done in exponential non-deterministic time. We then verify in EXPSPACE that there exists a quasi-model for ϕ in \mathcal{K} under \mathcal{S} and a . Since $\text{NEXPSPACE} = \text{EXPSPACE}$, this method runs in EXPSPACE. \square \square

A.3 Removing Negation

We first establish the complexity bounds for the non-probabilistic case, that is, entailment of *temporal queries* (TQs), which are TPQs without probability operators, from *temporal knowledge bases* (TKBs), which are TPKBs without probabilities different from 1.

For the entailment of TPQs without probability operators, i.e., positive TQs, complexity bounds are known from [8], or can be shown using standard techniques.

Lemma 13. *Entailment of positive TQs is in P wrt. data complexity and NP complete wrt. combined complexity, even if $\mathbf{N}_{\text{Crig}} \neq \emptyset$ and $\mathbf{N}_{\text{Rrig}} \neq \emptyset$.*

Proof. [8] show that query entailment is in P data complexity, even if $\mathbf{N}_{\text{Rrig}} \neq \emptyset$. The only remaining case is the combined complexity for \mathcal{EL} TKBs.

We describe an NP procedure for a given positive TQ ϕ and \mathcal{EL} TKB $\langle \mathcal{T}, (\mathcal{A}_i)_{i \in \llbracket 1, n \rrbracket} \rangle$. For $X \in \mathbf{N}_{\text{C}} \cup \mathbf{N}_{\text{R}}$, denote by $X^{(i)}$ the name X if $X \in \mathbf{N}_{\text{rig}}$, and a fresh name X^i if $X \notin \mathbf{N}_{\text{rig}}$, and for a given axiom/assertion/query α , denote by $\alpha^{(i)}$ the result of replacing every name X in α by $X^{(i)}$. Define an atemporal KB $\mathcal{K}' = \{\mathcal{T}', \mathcal{A}'\}$ based on the TKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{i \in \llbracket 1, n \rrbracket} \rangle$ by $\mathcal{T}' = \{\alpha^{(i)} \mid \alpha \in \mathcal{T}, i \in \llbracket 1, n+1 \rrbracket\}$ and $\mathcal{A}' = \{\alpha^{(i)} \mid \alpha \in \mathcal{A}_i, i \in \llbracket 1, n+1 \rrbracket\}$. \mathcal{K}' is polynomial in \mathcal{K} , and one can show that for any axiom/assertion/CQ α and $i \in \llbracket 1, n+1 \rrbracket$, we have $\mathcal{K}, i \models \alpha$ iff $\mathcal{K}' \models \alpha^{(i)}$ [10].

In order to decide entailment of a TQ ϕ , we guess a certificate that assigns to each pair (i, ψ) of a time point $i \in \llbracket 1, n+n_t \rrbracket$ and a CQ ψ occurring in ϕ a truth value, and, in case **true** is assigned to such a pair (i, ψ) , a certificate for the entailment of ψ at i (such a certificate exists since entailment of CQs is in NP wrt. combined complexity). For any time point after n , the entailment of a CQ solely depends on the rigid names. Therefore, for every CQ q in ϕ , if $\mathcal{K}, n+1 \models q$, then $\mathcal{K}, n+i \models q$ for all $i > 1$. Based on the guessed truth-assignment of CQs, we can now evaluate the entailment of ϕ as in the propositional case, which for LTL-formulae without negation symbols can be done in P [11]. As this certificate can be guessed and verified in non-deterministic polynomial time, we obtain an NP-upper bound. \square \square

The proof of Theorem 7 further depends on the following lemma, which limits the time points we have to consider explicitly.

Lemma 14. *Let ϕ be a TPQ, $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{i \in \llbracket 1, n \rrbracket} \rangle$ a TKB and n_t be the maximal nesting depth of temporal operators in ϕ . Then, for every $i > n + n_t$, $\mathcal{K}, i \models \phi$ iff $\mathcal{K}, n + n_t + 1 \models \phi$.*

Proof. We do the proof by structural induction on ϕ , and distinguish the cases based on the structure of ϕ .

1. If ϕ is a CQ, note that the only way in which \mathcal{K} restricts its models for time points after n is via its rigid names. Therefore, we have for all $i > n$, $\mathcal{K}, i \models \phi$ iff $\mathcal{K}, n + 1 \models \phi$.
2. If ϕ is of one of the forms $\psi_1 \wedge \psi_2$ and $\psi_1 \vee \psi_2$, the hypothesis follows by direct application of the inductive hypothesis.
3. If ϕ is of the form $\bigcirc^- \psi$, we have by inductive hypothesis that for all $i > n + n_t - 1$, $\mathcal{K}, n + n_t + 1 \models \psi$ iff $\mathcal{K}, i \models \psi$ iff $\mathcal{K}, i + 1 \models \bigcirc^- \psi$ iff $\mathcal{K}, j \models \phi$ for all $j > n + n_t$.
4. If ϕ is of one of the forms $\bigcirc \psi$, $\diamond \psi$, $\diamond^- \psi$, $\square \psi$, $\square^- \psi$, $\psi_1 \mathcal{U} \psi_2$ or $\psi_1 \mathcal{S} \psi_2$, we note that by inductive hypothesis, for all $i > n + n_t - 1$ $\mathcal{K}, i \models \psi$ (ψ_1, ψ_2) iff $\mathcal{K}, n + n_t \models \psi$ (ψ_1, ψ_2), which implies $\mathcal{K}, i \models \phi$ iff $\mathcal{K}, n + n_t + 1 \models \psi$ for all $i > n + n_t - 1$, and consequently also $\mathcal{K}, i \models \phi$ iff $\mathcal{K}, n + n_t + 1 \models \phi$ for all $i > n + n_t$. \square

\square

We can now provide the upper bounds stated in Theorem 7. A central technique used for this is to flatten TPQs using an abstraction of the probability expressions $\mathsf{P}_{\geq p}(\psi)$ occurring in the query. We identify each such expression with the assertion $A_{p,\psi}(a)$, where $A_{p,\psi}$ is fresh, which we add to the ipABox \mathcal{A}_i once we established that $\mathsf{P}_{\geq p}(\psi)$ is entailed at i . To capture this abstraction in a given TPQ ψ , we denote by ψ_f the result of replacing every outermost sub-query in ψ of the form $\mathsf{P}_{\geq p}(\psi)$ with $\exists x. A_{p,\psi}(x)$.

Lemma 15. *Entailment of TPQs from \mathcal{EL} - and DL-Lite-TPKBs is in PP wrt. data complexity, even if $\mathsf{N}_{\text{Rig}} \neq \emptyset$. It is in PP^{NP} wrt. combined complexity if the nesting depth of probability-operators in the query is bounded, and otherwise in P^{PP} .*

Proof. Before we consider nested probability operators, we consider the basic case of simple TPQs of the form $\mathsf{P}_{\geq p}(\phi)$, where ϕ does not contain any probability operators. Entailment of such a TPQ can be decided by checking for which possible world $w \in \Omega_{\mathcal{K}}$, $w, i \models \phi$, and then summing the probabilities of these worlds. This can be implemented by a probabilistic Turing machine (which uses an NP-oracle in the case of the combined complexities), which constructs a single possible world $w = (\mathcal{A}'_i)_{i \in \llbracket 1, n \rrbracket}$ on each branch, while taking care that the probabilities of the possible worlds are reflected by the probabilities in the Turing machine. For each $i \in \llbracket 1, n \rrbracket$ and $\alpha : p \in \mathcal{A}_i$, the machine adds α to \mathcal{A}'_i on b_1 succeeding branches, and does not add α to \mathcal{A}'_i on b_2 succeeding branches, where $\frac{b_1}{b_1 + b_2} = p$. After all axioms are processed, accept if $\langle \mathcal{T}, (\mathcal{A}'_i)_{i \in \llbracket 1, n \rrbracket} \rangle, i \models \phi$, which can be decided in P data complexity and NP combined complexity. By adding further dummy states to the Turing machine, we can ensure that the machine accepts at least half of its computation paths iff $\mathcal{K} \models \mathsf{P}_{\geq p}(\phi)$, so that entailment of the simple TPQ ϕ is decided in PP data complexity and PP^{NP} combined complexity.

To decide entailment of TPQS that contain several probability operators, we proceed in k rounds, where k is the maximal nesting depth of probability operators in ϕ , and test in each round for the entailment of probabilistic sub-queries at different time points. Let n_t denote the maximal nesting depth of temporal operators in ϕ . It can be shown that we have to consider only the

first $n + n_t$ time points. In each round $r \in \llbracket 1, k \rrbracket$, we iterate over all subformulae in ϕ that are of the form $\mathsf{P}_{\geq p}(\psi)$, where ψ contains at most $r - 1$ nestings of probability operators, and over all timepoints $i \in \llbracket 1, n + n_t + 1 \rrbracket$, and decide whether $\mathcal{K}, i \models \mathsf{P}_{\geq p}(\psi_f)$. If $\mathcal{K}, i \models \mathsf{P}_{\geq p}(\psi_f)$, we add $A_{p,\psi}(a)$ to \mathcal{A}_i . In the last round, we processed all probability operators, and decide whether $\mathcal{K} \models \mathsf{P}_{\geq 1}(\phi_k)$. Provided the nesting depth of probability operators is bounded, (as is always the case for data complexity), we can now use the fact that PP (and therefore also PP^{NP}) is closed under k -round polynomial truth table reductions [19]. These are defined as a sequence of k sets of polynomially many polynomial truth-table reductions, where k is a constant, and each truth-table reduction only depends on the input and the results of previous rounds. If the nesting-depth of probability operators is bounded, the above procedure can be described by such a reduction, and we obtain the PP and PP^{NP} upper bounds. Regarding the combined complexity with unbounded nesting of probability operators, we note that the above procedure can be implemented by a polynomial Turing machine that decides entailment of simple TPQs using a PP^{NP} oracle, so that we obtain a $\text{P}^{\text{PP}^{\text{NP}}}$ upper bound. Now, using Toda's result that $\text{PP}^{\text{PH}} \subseteq \text{P}^{\text{PP}}$ [30], we can internalise all calls to the PP^{NP} oracle in a P^{PP} machine, so that we obtain a P^{PP} upper bound for the combined complexity without bound on the nesting-depth of probability operators. \square

For the data complexity, our upper bound is matched by PP-hardness of the atemporal case [23]. We could not find a lower bound for the combined complexity in the literature for our precise setting (ipABoxes or tuple-independent databases). We therefore provide a proof for it here.

Lemma 16. *Entailment of TPQs from TPKBs is PP^{NP} -hard.*

Proof. We only need to provide a lower bound for the combined complexity. We do the proof by reduction of the PP^{NP} complete problem $\text{M}\exists\text{CNF3}$: given a QBF-formula of the form $\phi = \exists x_1, \dots, x_n. \phi'$, where ϕ' is a CNF3-formula over the variables $\{x_1, \dots, x_n, y_1, \dots, y_m\}$ with clauses $\{c_1, \dots, c_o\}$, decide whether at least half of assignments of truth values to the variables y_1, \dots, y_m make ϕ true [31]. As it turns out, we only need a single ipABox for this. The ipABox contains for every variable $x_i, i \in \llbracket 1, n \rrbracket$ the assertions $\mathsf{B}(x_i^+)$ and $\mathsf{B}(x_i^-)$, and for every variable $y_i, i \in \llbracket 1, m \rrbracket$ the assertions $\mathsf{B}_i^+(y_i^+) : 0.5$ and $\mathsf{B}_i^-(y_i^-) : 0.5$. Intuitively, $\mathsf{B}_i^+(y_i^+)$ is entailed in a possible world that corresponds to an assignment of true to the variable y_i , while $\mathsf{B}_i^-(y_i^-)$ is entailed in a possible world that corresponds to an assignment of false to the variable y_i . Since all probabilities are independent, we will have worlds that correspond to “invalid variable assignments”, in the sense that they either do not assign a truth value to every variable, or multiple truth values. We will take care of this later. We use the TBox axioms $B_i^+ \sqsubseteq B_i$, $B_i^- \sqsubseteq B_i$, $B_i^+ \sqsubseteq B^+$, $B_i^- \sqsubseteq B^-$, $B^+ \sqsubseteq B$ and $B^- \sqsubseteq B$ to abstract away from the specific assignment if needed.

For every literal l , denote by $v(l)$ the variable in l . For every clause $c_j = l_1 \vee l_2 \vee l_3, c \in \llbracket 1, o \rrbracket$, and truth valuation π that makes c_j true, add the assertions

$$M(c_j, \pi), M(\pi, l'_1), M(\pi, l'_2), M(\pi, l'_3),$$

where for $i \in \llbracket 1, 3 \rrbracket$, $l'_i = v(l_i)$ if $\pi(l_i) = \mathbf{true}$, and $l'_i = \neg v(l_i)$ if $\pi(l_i) = \mathbf{false}$. As last assertion, we add $\mathsf{H}(a) : 0.5$, which only serves the purpose of being satisfied in at least half of the possible worlds.

Our CQ is now composed of three queries q_1, q_2 and q_3 defined next. The query

$$q_1 = \exists y_1, \dots, y_m : B_1(y_1) \dots B_m(y_m)$$

is entailed in every possible world which assigns a truth value to each variable $y_i, i \in \llbracket 1, m \rrbracket$. The query

$$q_2 = \exists y : B^+(y) \wedge B^-(y)$$

is entailed in the possible worlds that assign two truth values to some variable y . Finally, the query

$$q_3 = \exists x_1, \dots, x_n, y_1, \dots, y_m, z_1, \dots, z_o : \bigwedge_{i \in \llbracket 1, o \rrbracket} \tau(c_i),$$

where for $c_i = l_1 \vee l_2 \vee l_3$,

$$\begin{aligned} \tau(c_i) = & M(c_i, z_i), M(z_i, v(l_1)), M(z_i, v(l_2)), M(z_i, v(l_3)) \\ & \wedge \bigwedge_{i \in \llbracket 1, 3 \rrbracket} B(v(l_i)), \end{aligned}$$

is satisfied in all possible worlds that correspond to an assignment that make ϕ true. q_3 can only be entailed in a possible world in which q_1 is also entailed (otherwise, we lack variables for some of the clauses). The query $(q_1 \wedge q_2) \vee (H(c) \wedge q_2)$ is entailed in (1) all possible worlds that correspond to an assignment that is complete but assigns to at least one variable two values and (2) half of the possible worlds that correspond to assignments that are both incomplete and assign two values to a variable. Due to symmetry, this query is thus entailed in exactly half of those possible worlds that do not correspond to a valid variable assignment. Consequently, the query

$$q = (q_1 \wedge q_2) \vee (H(c) \wedge q_2) \vee q_3$$

is entailed in more than half of all possible worlds iff ϕ is satisfied for more than half of its valid assignments, so that $P_{\geq 0.5}(q)$ is entailed iff ϕ is satisfied by at least half of the assignments. Note furthermore that both the TPKB and the TPQ are polynomial in the input, so that we obtain the PP^{NP} lower bound. \square \square